



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR
LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA
GENERACIÓN A LA QUE ESTOS PERTENECEN**

Abner Eleazar Cojón Conde

Asesorado por el Ing. David Estuardo Morales Ajcót

Guatemala, noviembre de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR
LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA
GENERACIÓN A LA QUE ESTOS PERTENECEN**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ABNER ELEAZAR COJÓN CONDE

ASESORADO POR EL ING. DAVID ESTUARDO MORALES AJCOT

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Luis Fernando Espino Barrios
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
EXAMINADOR	Ing. William Estuardo Escobar Argueta
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA GENERACIÓN A LA QUE ESTOS PERTENECEN

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha de agosto de 2018.



Abner Eleazar Cojón Conde



Guatemala 27 de junio 2019.

Ingeniero

Carlos Alfredo Azurdia Morales

Coordinador del Área de Trabajos de Graduación

Respetable Ingeniero Azurdia:

Por este medio informo que he revisado y aprobado el trabajo de investigación titulado: "**METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA GENERACIÓN A LA QUE ÉSTOS PERTENECEN**", desarrollado por el estudiante **Abner Eleazar Cojón Conde**, quien se identifica con el número de carné **200915713**, ya que considero que cumple con los requisitos establecidos, por lo que el autor y mi persona somos responsables del contenido y conclusiones del mismo.

Agradeciendo su atención a la presente.

Atentamente.

Ing. David Estuardo Morales Ajcort

Asesor

No. Colegiado 10993

David Estuardo Morales Ajcort
Ingeniero en Ciencias y Sistemas
Colegiado No. 10933



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 17 de julio de 2019

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **ABNER ELEAZAR COJÓN CONDE** con carné **200915713** y CUI **1841 75496 0401** titulado **"METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA GENERACIÓN A LA QUE ÉSTOS PERTENECEN"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS
TEL: 24188000 Ext. 1534

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación, **"METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA GENERACIÓN A LA QUE ESTOS PERTENECEN"** realizado por el estudiante, ABNER ELEAZAR COJÓN CONDE, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

MSc. Ing. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 07 de noviembre de 2019



Ref. DTG.528.2019

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **METODOLOGÍAS DE DESARROLLO DE SOFTWARE UTILIZADAS POR LÍDERES DE PROYECTOS DE TECNOLOGÍA Y SU RELACIÓN CON LA GENERACIÓN A LA QUE ESTOS PERTENECEN**, presentado por el estudiante universitario: **Abner Eleazar Cojón Conde**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRESIÓN.



DECANO
FACULTAD DE INGENIERÍA

Inga. Aurelia Anabela Cordova Estrada
Decana

Guatemala, noviembre de 2019

/cc

ACTO QUE DEDICO A:

Dios	Porque toda gloria es para Él.
Mis padres	Oscar Cojón y Letty Conde de Cojón (q.e.p.d), por ser mi mayor inspiración.
Mis hermanos	Caleb y Keyla Cojón, por ser mis mejores amigos.
Mi sobrina	Yamileth Cojón, por ser una de las razones de mi felicidad.
Mi amiga especial	Sarah Fabiola Sosa Perdomo (q.e.p.d), por el amor e inspiración.
Mis amigos	Quienes ya se han graduado, han sido motivación y a quienes están en formación para motivarlos a luchar para cumplir sus metas.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser la casa de estudios en donde formé gran parte del carácter profesional.
Facultad de Ingeniería	Por ser una importante influencia en mi carrera.
Mis amigos de la Facultad	Gustavo Palacios, Cristina Roncal, Bryam de León, Félix García, Andrea Chavarría, Fernando García, Christian González y Ricardo Barrios, por su importante apoyo en este trayecto.
Ing. Raúl Mijangos, Ing. David Morales, Ing. Hugo Rivera	Por su amistad y apoyo profesional en la fase final de esta carrera.
Mis amigos	Mónica Rosales, Sanders Orozco, Daniela González, Andrea Estrada, Giessel Estrada, Lucía Oliva, Karen García y Anny Melgar, por su valiosa amistad y apoyo espiritual.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	VII
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN	XV
1. METODOLOGÍAS DE DESARROLLO DE SOFTWARE Y ANÁLISIS GENERACIONAL.....	1
1.1. Definición de metodología de desarrollo de software	1
1.2. Clasificación de las metodologías de desarrollo de software	1
1.2.1. Metodologías clásicas	1
1.2.1.1. Modelo en cascada.....	2
1.2.1.1.1. Entregables por cada etapa, incluye fase de instalación	3
1.2.1.2. Modelo en espiral	5
1.2.1.2.1. Entregables por cada etapa	6
1.2.1.3. Modelo de prototipos evolutivos	7
1.2.1.3.1. Entregables por cada etapa	9
1.2.2. Metodologías ágiles.....	9
1.2.2.1. Principios del manifiesto ágil.....	10
1.2.2.2. Scrum	10

	1.2.2.2.1.	Entregables en cada etapa	11
	1.2.2.3.	Programación extrema	12
	1.2.2.3.1.	Entregables por cada etapa	13
	1.2.2.4.	Kanban	13
	1.2.2.4.1.	Entregables en cada etapa	15
1.3.		Principios básicos de contratos de operaciones para proyectos de desarrollo de software.....	15
	1.3.1.	Clasificación de tipos de contratos de operaciones.....	16
	1.3.1.1.	Contratos de precio fijo	16
	1.3.1.2.	Contrato por costes reembolsables	16
	1.3.1.3.	Contrato por tiempos, materiales y servicios	16
	1.3.1.4.	Estructura básica de un contrato.....	17
1.4.		Clasificación generacional.....	17
	1.4.1.	Generación Baby Boomers	18
	1.4.2.	Generación X	18
	1.4.3.	Generación Y (<i>millennials</i>)	19
	1.4.4.	Generación Z (<i>centennials</i>)	20
	1.4.5.	Comparación de tendencias e influencias tecnológicas	22
	1.4.6.	Comparación de metodologías de su formación académica	23
2.		ANÁLISIS DE SITUACIÓN ACTUAL Y PROPUESTA DE SOLUCIÓN ..	25
	2.1.	Definición del problema.....	25

2.2.	Propuesta de solución al problema	26
2.2.1.	Estructura de encuesta y detalle del objetivo de cada elemento	26
3.	ANÁLISIS DE RESULTADOS	29
3.1.	Tabulación de datos	30
3.1.1.	Resultados obtenidos con respecto a la edad de los encuestados.....	30
3.1.2.	Resultados obtenidos con respecto al rol de participación de los entrevistados en los proyectos que ejecutan.....	31
3.1.3.	Metodologías de desarrollo en la etapa universitaria	33
3.1.4.	Mercado objetivo de los proyectos en los que labora.....	34
3.1.5.	Población económicamente activa	35
3.1.6.	Metodologías consideradas idóneas para el mercado guatemalteco en estudiantes	36
3.1.7.	Metodologías utilizadas actualmente en el mercado laboral un mes atrás	37
3.1.8.	Metodologías utilizadas a la fecha en el mercado laboral.....	38
3.2.	Clasificación de líderes según su criterio de decisión	39
3.3.	Análisis de resultados por generación	42
3.3.1.	Generación X.....	42
3.3.2.	Generación Y	43
	CONCLUSIONES	45
	RECOMENDACIONES.....	47

BIBLIOGRAFÍA.....49

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Etapas del ciclo de vida de desarrollo, modelo en cascada	3
2.	Etapas del ciclo de vida, modelo en espiral	7
3.	Resultados. Fecha de nacimiento	30
4.	Resultados. Rol como interesado en el proyecto	31
5.	Resultados. Metodologías de desarrollo aprendidas en etapa universitaria.....	33
6.	Resultados. Mercado objetivo de los proyectos en los que trabaja.....	34
7.	Resultados. Segmentación de población económicamente activa.....	35
8.	Resultados. Metodologías idóneas para el mercado guatemalteco	36
9.	Resultados. Metodologías usadas hasta un mes atrás de la encuesta.....	37
10.	Resultados. Metodologías usadas en la fecha de encuesta	38
11.	Clasificación de metodologías elegidas por generación (cifra porcentual)	39
12.	Características preferidas de las metodologías (cifra porcentual).....	40
13.	Criterios de aceptación por metodología (cifra porcentual)	41

TABLAS

I.	Comparación de tendencias e influencias tecnológicas	22
II.	Comparación de metodologías de su formación académica.....	23
III.	Clasificación de líderes según su generación y criterio de decisión.....	42

GLOSARIO

BDD	Base de Datos. Colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.
Desarrollador	Profesional de la informática que se dedica a la programación de computadoras.
DFD	Diagrama de Flujo de Datos. Herramienta gráfica que muestra de forma ordenada el flujo de datos.
CRC	<i>Class-Responsibility-Collaboration</i> . Tarjeta de inventario de responsabilidad, utilizada en metodología de programación extrema.
Fichero	Es un conjunto de bits que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.
Hardware	Parte física de un ordenador o sistema informático.
Informática	Procesamiento automático de información mediante dispositivos electrónicos y sistemas computacionales.

Interfaz de usuario	Medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.
Librería	Es un programa que se integra dentro de otro para dotarlo de funcionalidad y lograr sus propósitos.
Módulo	Una porción de un programa informático.
Paquete informático	Es un programa que permite a los usuarios realizar diversos trabajos en la automatización de tareas complicadas.
<i>Product owner</i>	Cliente. Rol entre los interesados del proyecto en la metodología Scrum, encargado de priorizar los objetivos.
Prueba unitaria	Método que se puede invocar desde el código programado para que este sea probado, se compara entre resultado real y esperado.
<i>Script</i>	Conjunto ordenado de instrucciones en determinado lenguaje de programación.
Sistema informático	Es un sistema que permite almacenar y procesar información. Es el conjunto de partes interrelacionadas: hardware, software y personal informático.

Software

Conjunto de programas y rutinas que permiten a la computadora realizar determinada tarea.

RESUMEN

A nivel mundial la tecnología evoluciona y marca un ritmo muy apresurado en el manejo de la información. A partir de eso se evidencia la necesidad de desarrollar sistemas que cumplan ese manejo adecuado de la información, en contexto de tiempo y seguridad.

En Guatemala se ha incrementado de forma considerable la demanda de desarrollo e implementación de sistemas informáticos y esto, a su vez, crea la necesidad de investigación y especialización para el uso de metodologías de desarrollo de software.

En las diferentes casas de estudios universitarios se enseñan metodologías de desarrollo de software sin profundizar en ellas o dar opción a que los estudiantes se especialicen. En general se imparten metodologías clásicas y ágiles por igual.

Se presenta una investigación que busca relacionar las metodologías de desarrollo de software elegidas por los líderes de proyectos y la generación a la que estos pertenecen, por medio de un análisis generacional, entrevistas y encuestas. Esto para obtener datos y tendencias, las cuales permitan generar una herramienta útil para todo profesional o aspirante del desarrollo de software, para tomar decisiones y especializarse en metodologías preferidas en el mercado.

OBJETIVOS

General

Demostrar la relación existente entre las metodologías de desarrollo de software elegidas por los líderes de proyectos y la generación a la que estos pertenecen.

Específicos

1. Identificar de forma precisa los entregables correspondientes a cada metodología de desarrollo de software.
2. Identificar la tendencia en la elección de metodologías ágiles con principios de las metodologías clásicas.
3. Identificar y detallar los criterios de decisión de los líderes de proyectos sobre la base de la generación en la que se clasifican.

INTRODUCCIÓN

La presente investigación trata de los criterios de decisión de líderes de proyectos de desarrollo de software, respecto a las metodologías que eligen para la gestión de sus proyectos y la relación que existe con la generación a la que estos pertenecen.

Para identificar las tendencias existentes en la toma de decisiones se realizará análisis y clasificación generacional y estudio de la evolución tecnológica, esto a través de la línea de tiempo de formación académica y profesional de los líderes de proyectos.

El principal punto de interés es brindar a los profesionales del área de ciencias y sistemas información que agregue valor profesional según el área en la que se desenvuelvan, siendo ellos estudiantes, desarrolladores, líderes de proyecto o consultores.

El estudio de las distintas ramas de las ciencias implicadas se realizará desde sus fundamentos hasta los puntos que permitan relacionar una ciencia con otra, con base en el tema de investigación. Se utilizarán referencias bibliográficas y electrónicas.

Los datos de la experiencia de los líderes de proyectos de desarrollo de software se obtendrán por medio de encuestas y entrevistas, las cuales estarán diseñadas para cumplir con el objetivo de clasificar los criterios de decisión e identificar la relación con la generación a la que los líderes pertenecen.

1. METODOLOGÍAS DE DESARROLLO DE SOFTWARE Y ANÁLISIS GENERACIONAL

1.1. Definición de metodología de desarrollo de software

Término también conocido como ingeniería del software, trata del conjunto de procesos estructurados que se siguen para desarrollar software y brindar productos como soluciones a medida de un cliente o sector de mercado. Involucra comunicación, manipulación e intercambio de información de manera estratégica, entre las partes involucradas.

1.2. Clasificación de las metodologías de desarrollo de software

La clasificación de las metodologías de desarrollo de software se ha realizado en dos grupos. Entre los criterios para la clasificación los principales son: documentación inicial o planeación del proyecto, prácticas que se realizan en cada una de las fases del ciclo de vida del proyecto y los entregables al finalizar el ciclo de vida del proyecto.

1.2.1. Metodologías clásicas

Clasificación que también puede ser conocida como tradicional. Las metodologías que se describen en esta clasificación son las primeras metodologías definidas e implementadas en la ingeniería de software, por lo cual marcan una base importante para la creación de las metodologías ágiles. Una de las principales características generales de las metodologías clásicas es

que requieren documentación más estricta desde la negociación hasta la finalización del ciclo de vida del proyecto.

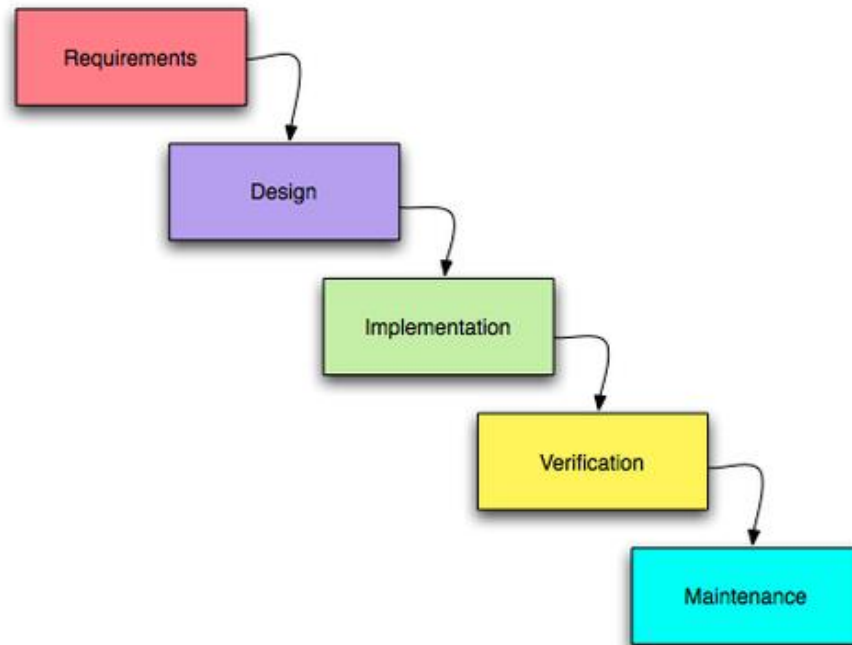
1.2.1.1. Modelo en cascada

Representa un proceso lineal y secuencial, en el cual el desarrollo de software se debe realizar en etapas ejecutadas una tras otra, partiendo de la estricta finalización de una etapa para poder ejecutar la siguiente. De esa representación se origina su nombre, pues el flujo es conforme una cascada.

Se supone que este modelo de desarrollo se originó en la industria y construcción, donde los cambios no planificados son más caros y su implementación es más complicada, en comparación con una solución de software. Este modelo de desarrollo se adaptó a la ingeniería de software dados los buenos resultados en otras áreas de aplicación y pocas opciones específicamente al área informática, esto aproximadamente en la década de los 70's.

Las etapas del ciclo de vida de un proyecto con la utilización del modelo en cascada son: requisitos (análisis), diseño, implementación, verificación y mantenimiento (ver figura 1).

Figura 1. **Etapas del ciclo de vida de desarrollo, modelo en cascada**



Fuente: NYMAND, Kasper. *Agile vs. Cascada (Parte 1 de 5). ¿Qué es la metodología en cascada?* www.medium.com. Consulta: agosto de 2018.

1.2.1.1.1. Entregables por cada etapa, incluye fase de instalación

- Requisitos (análisis).
 - Captura de requerimientos.
 - Análisis de necesidades o sistema actual.
 - Nuevos requerimientos de usuario.
 - Documento descriptivo del sistema propuesto.
 - Especificaciones del sistema.
 - Documento de DFD's.
 - Requisitos de BDD.
 - Requisitos de infraestructura y telecomunicaciones.
 - Requisitos de hardware.

- Plan de pruebas de integración.
- Diseño.
 - Descripción detallada del contenido del sistema.
 - Módulos, objetos, programas.
 - Ficheros y base de datos.
 - Diccionario de datos.
 - Procedimientos.
 - Estimación de carga del sistema y tiempos de respuesta esperados.
 - Interfaces de usuario.
 - Estándares de programación y diseños utilizados y recomendados.
 - Técnicas de implementación recomendadas.
 - Codificación local.
 - Compra de paquetes y librerías.
 - Plan de pruebas de integración con programas.
- Implementación (desarrollo).
 - Documentos del diseño final del sistema y cada programa integrado.
 - DFD del sistema y programas integrados.
 - Descripción con lógica de cada programa integrado.
 - Descripción de entradas y salidas del sistema.
 - Cadenas y secuencias de ejecución si es necesario, especialmente en *scripts* de BDD.
 - Resultado de las pruebas de cada proceso.
 - Resultado de pruebas de cada módulo.
 - Resultado de pruebas de integración.
 - Manual de usuario final.
- Verificación (pruebas).

- Plan finalizado de pruebas del sistema.
- Informe de los resultados de pruebas, detalle de resultados esperados y obtenidos.
- Plan de corrección ante resultados no esperados.
- Informe de resultados de las pruebas a la documentación.
- Instalación.
 - Detalle de plan de contingencia ante sobrecarga, caída y recuperación del sistema.
 - Plan de revisión post instalación.
 - Carta de aceptación del sistema.
- Mantenimiento.
 - Listado de fallas detectadas en el sistema.
 - Detalle de los cambios realizados en el sistema como corrección o mejora.
 - Actas correspondientes a las revisiones regulares del sistema.
 - Control de aceptación de niveles de soporte.

1.2.1.2. Modelo en espiral

Representa un modelo de desarrollo de software evolutivo con naturaleza iterativa, construyendo prototipos en fases más cortas, pero con las características del modelo lineal secuencial. Los prototipos desarrollados son una serie de versiones incrementales del software. Las primeras iteraciones pueden ser desde un diseño simple en papel hasta prototipos no funcionales, diseño de interfaces de usuario. A medida que aumentan las iteraciones, las versiones son más completas y se muestra una solución madura del sistema diseñado. Este modelo crea la necesidad de abrir la comunicación entre el cliente y desarrollador desde las primeras etapas del ciclo de vida del software.

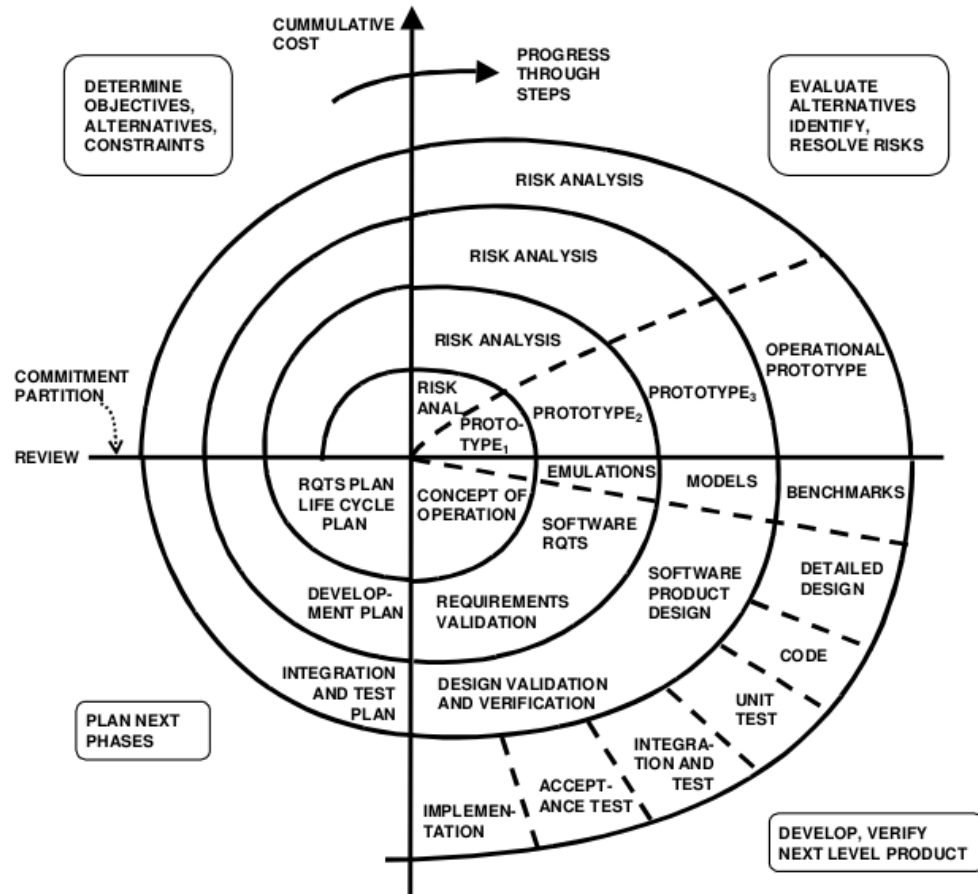
Las etapas por cada iteración de este modelo son: comunicación con el cliente, planificación, análisis de riesgos, ingeniería (desarrollo), evaluación del cliente y construcción, y entrega.

El modelo plantea un ciclo ideal en el que cada iteración complete las siguientes etapas de madurez: desarrollo de los conceptos, desarrollo del nuevo producto, mejora del producto y mantenimiento del producto (ver figura 2).

1.2.1.2.1. Entregables por cada etapa

- Planificación.
 - Documento detalle de objetivos generales y específicos.
 - Plan de gestión del proyecto.
 - Documento de identificación de riesgos.
- Análisis de riesgos.
 - Documentación con detalle de pasos para reducción de los riesgos identificados.
- Ingeniería (desarrollo).
 - Prototipo.
- Evaluación del cliente.
 - Documento de avances en el prototipo.
 - Documento de correcciones del ciclo anterior.
- Construcción y entrega.
 - Informe general de avances de todo el proyecto.
 - Carta de aceptación del sistema.

Figura 2. **Etapas del ciclo de vida, modelo en espiral**



Fuente: GURENDO, Dmitry. *Software Development Life Cycle (SLDC). Spiral Model*.
www.xbsoftware.com. Consulta: agosto de 2018.

1.2.1.3. **Modelo de prototipos evolutivos**

Modelo de desarrollo evolutivo, parte de la premisa que los requerimientos no son conocidos completamente al inicio del proyecto. Es reconocido y utilizado para dar al usuario una vista preliminar de las interfaces o funcionamiento limitado del sistema propuesto. Es tomado en muchas ocasiones como prueba y error, ya que si el usuario no aprueba un prototipo significa que ha fallado el diseño y la definición de este y debe ser corregido para obtener la aprobación. Una característica importante es el tiempo en el que

se desarrolla el prototipo, debe ser un tiempo corto, pero suficiente para modelar la interfaz o funcionamiento de una parte del sistema.

La construcción de un prototipo no garantiza la mejor calidad de software, pero sí permite al usuario y parte encargada del desarrollo aclarar si este cumple con los requerimientos y necesidades o si deberán considerarse más detalles y mejoras.

Es de alta importancia considerar que el lenguaje de programación a utilizar en la etapa de desarrollo debería ser adaptable a los prototipos, es decir, que permita experimentar. Esto para optimizar el tiempo una vez el prototipo ya se haya aprobado y hacer uso de prototipos evolutivos y no desechables.

Los prototipos evolutivos son los que se adaptan al desarrollo de software, por tanto, el desarrollo de software en un ciclo será el prototipo del siguiente, esto denota optimización de tiempo, seguimiento y estandarización en la lógica de la solución.

Los prototipos desechables generalmente son interfaces de usuario sin modelación de funcionamiento con la lógica de negocio y contruidos con herramientas que no se adaptan al entorno de desarrollo de software, razón suficiente para ser desechables.

Las etapas del modelo son: recolección y refinamiento de requisitos, modelado, construcción del prototipo, evaluación del prototipo por el cliente, refinamiento del prototipo y producto de ingeniería.

1.2.1.3.1. Entregables por cada etapa

- Recolección y refinamiento de requisitos.
 - Documento con detalle de requisitos del sistema.
- Modelado.
 - Diseño rápido del prototipo, puede ser tan informal como algo a papel y lápiz.
- Construcción del prototipo.
 - Prototipo.
- Evaluación del prototipo por el cliente.
 - Carta de aceptación o rechazo del prototipo.
- Refinamiento del prototipo.
 - Prototipo con la aplicación de las mejoras o correcciones.
- Producto de ingeniería.
 - Resultado del desarrollo de software a partir del prototipo aprobado.
 - Carta de aceptación del producto.

1.2.2. Metodologías ágiles

Métodos de desarrollo de software enfocados en la evolución conjunta entre las necesidades y las soluciones, por medio de la estrecha colaboración entre los equipos multidisciplinarios.

1.2.2.1. Principios del manifiesto ágil

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para la continuación ajustar y perfeccionar su comportamiento en consecuencia.¹

1.2.2.2. Scrum

Conjunto de buenas prácticas aplicadas de forma constante, que permiten trabajar en colaboración entre los equipos multidisciplinarios con el fin de

¹ MEDINILLA, A., GINÉ, A., GÓMEZ, E. *Principios del Manifiesto Ágil*.
<http://agilemanifesto.org/iso/es/principles.html>. Consulta: octubre 2018

obtener el mejor resultado del proyecto, en este contexto de desarrollo de software.

Se utiliza en proyectos de distintas dimensiones, pero su enfoque son proyectos grandes y de alta complejidad. Los requisitos cambiantes o con poca definición y la necesidad de resultados útiles en un corto tiempo, son factores que aumentan la complejidad de cualquier proyecto. La solución ante tal complejidad está basada en la priorización realizada por el *product owner*, según la necesidad de quien representa. Su proyección a innovación, competitividad, flexibilidad y alta productividad ha propiciado ser una de las metodologías más utilizadas.

El proyecto se debe ejecutar en ciclos temporales cortos, con una clara definición de la duración, llamados iteraciones. La definición del tiempo deberá ser en común acuerdo, basada en el análisis del tiempo de cada una de las siguientes actividades: planificación de la iteración, ejecución de la iteración, inspección y adaptación.

1.2.2.2.1. Entregables en cada etapa

- Planificación de la iteración.
 - Listado de requisitos priorizados.
 - Detalle de tareas necesarias para desarrollar los requisitos.
 - Estimación del esfuerzo.
- Ejecución de la iteración.
 - Informe hablado respecto a avances, obstáculos y nuevas necesidades para cubrir iteración, esto en cada reunión de sincronización por cada miembro del equipo.

- Minuta de reunión, necesaria solamente cuando hay cambios de alto impacto en los requisitos.
- Inspección y adaptación.
 - Presentación de los requisitos completados.
 - Producto evolucionado o nuevo módulo.
 - Detalle del incremento del producto respecto a la iteración anterior.
 - Detalle cronológico de las iteraciones pasadas que refleje el crecimiento del proyecto.

1.2.2.3. Programación extrema

Metodología basada en la simplicidad, la buena comunicación y código reutilizable. Es planteada como respuesta a la necesidad de herramientas que permitan mantener control en proyectos con requerimientos cambiantes y un alto riesgo técnico. Su principal enfoque es potenciar las relaciones interpersonales para el éxito del proyecto en general, pues se parte de la premisa “todo el software cambia”² y, si ese cambio no se puede prever, la buena comunicación entre los interesados del proyecto reduce la incapacidad de adaptarse al cambio.

Esta metodología hace uso de las prácticas basadas en el manifiesto ágil, pero llevadas al extremo. El uso del término “extremo” puede causar contradicción con otro de los objetivos de la metodología, crear un agradable ambiente de trabajo. Se promueve el trabajo en equipo en todas sus actividades, lo que permite a la persona asignada al desarrollo tener un proceso

² WELLS, Don. *Extreme Programming: A gentle introduction*.
<http://www.extremeprogramming.org/>. Consulta: octubre 2018.

de aprendizaje y adaptación en tiempo récord. A pesar de la rapidez y elevación en la curva de aprendizaje del equipo de desarrollo, se requiere un alto nivel profesional en ellos, dados los cambios de roles, alta interacción con el cliente y con énfasis especial en la programación en parejas.

Las etapas iterativas del ciclo de vida del software son: planificación, diseño, desarrollo y pruebas.

1.2.2.3.1. Entregables por cada etapa

- Planificación
 - Historias de usuario
 - Plan de rotaciones
 - Minutas de reuniones de seguimiento con detalle de correcciones
- Diseño
 - CRC
- Desarrollo
 - Presentación del producto en construcción
 - Plan de pruebas unitarias
 - Informe comparativo con versión anterior
- Pruebas
 - Informe de pruebas unitarias
 - Carta de aceptación

1.2.2.4. Kanban

Su contexto con la ingeniería de software la define como una metodología cuyo principal objetivo es la gestión general de cómo se van completando las tareas en uno o más proyectos. La herramienta utilizada para el seguimiento del progreso son tarjetas visuales en las que se identifica de forma única la tarea y

su responsable, las cuales se mueven en un tablero que representará cada uno de los pasos estrictamente necesarios para completar el ciclo de vida desde la identificación de una necesidad hasta la resolución de esta. Los estados básicos de las tareas en el tablero son: requerida, en progreso y finalizada.

El movimiento de las tareas en el tablero permite identificar las tareas que se realizan con mayor efectividad y las que causan cuellos de botella. Esta metodología tiene un enfoque realista, mantiene como prioridad limitar el trabajo en progreso, por la efectividad de la inversión de tiempo y recursos en el presente.

La implementación es bien aceptada, dado que uno de sus principios para su etapa inicial es mantener los procesos en el orden actual. Es mera estrategia, pues si se conoce y analiza el proceso actual, entonces se podrá identificar y proponer mejoras.

El seguimiento debe realizarse en circuitos de retroalimentación, iniciando con reuniones cortas donde el equipo de desarrollo reporte los avances de un día anterior y la proyección de avances al finalizar el día actual. Los resultados obtenidos en las revisiones internas se presentan al cliente como el progreso obtenido en la iteración.

Las etapas no han sido definidas de forma estricta, son más adaptables los ambientes con los que cuente el equipo desarrollo, las etapas ideales son: requerimientos, desarrollo, pruebas de equipo de desarrollo, pruebas de gestión de calidad y despliegue.

1.2.2.4.1. Entregables en cada etapa

- **Requerimientos**
 - Documento con detalle de la toma de requerimientos
 - Diseño de DFD's
 - Diseño de interfaces de usuario no funcionales
- **Desarrollo**
 - Producto en construcción
 - Plan de pruebas
- **Pruebas del equipo de desarrollo**
 - Informe de resultados de pruebas
- **Pruebas de gestión de calidad**
 - Informe de resultados de pruebas
 - Informe de tareas y su estado según la aceptación
- **Despliegue**
 - Producto parcial o finalizado (según iteración)
 - Carta de aceptación del producto

1.3. Principios básicos de contratos de operaciones para proyectos de desarrollo de software

En la gestión de proyectos es normal que se presente la necesidad de subcontratar parte del trabajo. Dado el enfoque de esta investigación, el rol en ese subcontrato puede ser ofreciendo un servicio o requiriendo este. Se presenta una clasificación de contratos de operaciones para proyectos de desarrollo de software para que la parte interesada del proyecto elija la opción que mejor se adapte a la metodología de desarrollo de software a utilizar.

1.3.1. Clasificación de tipos de contratos de operaciones

En esta sección se presenta la clasificación de los tipos de contratos de operaciones y el detalle de la estructura básica de un contrato.

1.3.1.1. Contratos de precio fijo

El cliente y contratista acuerdan un precio desde el inicio del proyecto, que no tendrá variación alguna. Las órdenes de compra son herramientas utilizadas cuando el pago se realiza en dos o más partes.

1.3.1.2. Contrato por costes reembolsables

El cliente acuerda con el contratista pagarle todos los costos reales en los que incurra durante el proyecto, más honorarios. El contratista debe tener especial cuidado en la facturación para no incurrir en pérdidas. Se requiere constante revisión por parte del cliente para garantizar que los costos realmente sean de beneficio al proyecto.

1.3.1.3. Contrato por tiempos, materiales y servicios

En este tipo de contrato se toman características de los dos anteriores, se fija el precio del servicio medido en horas y se toma como un componente variable, la cantidad de horas necesarias para completar el proyecto. Muy pocas veces se podrá realizar el estimado exacto de las horas de forma ajustada, pues existen factores externos que pueden afectar a ambas partes, estos impiden el cumplimiento si no se hace un estimado con algún tipo de holgura.

1.3.1.4. Estructura básica de un contrato

- Nombre: identificador de la operación y sus parámetros.
- Responsabilidades: descripción general de los requerimientos que la operación deberá satisfacer.
- Referencias cruzadas: números de referencia de los requerimientos funcionales del sistema y las herramientas que reflejen los procedimientos de forma atómica. Esto dependerá de la metodología de desarrollo elegida, en general se detalla identificador de casos de uso.
- Notas: toda la documentación relacionada al diseño, DFD's, interfaces de usuario, algoritmos, entre otros.
- Excepciones: descripción de las situaciones que se pueden presentar y que impiden el avance ideal en el proyecto. Debe detallarse la frecuencia con que se presentan tales situaciones.
- Salida: mensajes o registros que se envían fuera del sistema a desarrollar, este apartado en la mayoría de los proyectos queda vacío. Es tan importante el análisis de la posible ocurrencia de este tipo de salidas, pues deberá incluirse en el plan de integración.
- Precondiciones: detallan el escenario previo a ejecutar la operación, se debe prestar especial atención a este apartado pues la omisión puede ser de alto impacto para el cliente o el contratista.
- Postcondiciones: detalla el escenario tras completar la operación, puede incluir todos los resultados esperados del proyecto.

1.4. Clasificación generacional

En esta sección se realiza una clasificación de personas por el año en que nacieron, describiendo las principales características en formación académica, laboral y su interacción con la tecnología.

1.4.1. Generación Baby Boomers

Esta generación es llamada de esa forma debido a la gran cantidad (*boom*) de nacimientos que se dio durante el segundo y tercer cuarto del siglo 20. En Estados Unidos de América se reconoce dentro de esta generación a los nacidos entre 1946 y 1964³. Los *baby boomers* latinoamericanos tienen ligeras diferencias a los de Estados Unidos de América, esta generación vivió en carne propia las revoluciones, crisis financieras locales y globales. Debido a esa experiencia tienden a ser buenos administradores, prefieren adquirir bienes inmuebles en lugar de rentar.

Su período económico más productivo se dio en las décadas 60's y 70's, aunque algunos aún se encuentran trabajando, otros jubilados o en proceso de hacerlo. Entre sus actividades de ocio se encuentra todo lo relacionado con noticias, vistas en TV, escuchadas en radio o leídas en periódicos.

Su perfil personal muestra personas tradicionalistas, rígidas y estructuradas en todo aspecto de su vida. Laboralmente cuidan su trabajo y respetan todas las normas internas y externas a la organización.

1.4.2. Generación X

Esta generación es la que actualmente mueve al mundo a nivel económico y laboral. Se clasifican dentro de esta generación los nacidos entre 1965 y 1984⁴, hijos de los *baby boomers*. Se nombra esta generación debido a la complejidad de los *baby boomers* para entenderlos, así que tomaron como referencia la novela de Douglas Coupland titulada *Generación X, tales for an*

³ PEREZBOLDE, Guillermo. *Diferencias entre generaciones*. <https://www.merca20.com/conoce-las-diferencias-entre-millennials-genx-y-baby-boomers/3>. Consulta: octubre 2018.

⁴ Ibíd.

accelerate culture (publicada en 1991). Esta nueva generación marcó un rumbo más rebelde y menos tradicionalista.

Esta generación es la gran impulsora de la tecnología, pues nacieron en la transición del uso de aparatos eléctricos al uso de aparatos electrónicos. A pesar de su gusto por la tecnología, quienes marcan el inicio de esta generación presentan alguna indiferencia ante el uso de Internet y redes sociales.

A nivel laboral se denota una fuerte ambición por el crecimiento y reconocimiento dentro de las organizaciones, siempre aspirando a posiciones gerenciales y directivas, aun así son más colaborativos que los *baby boomers*, se adaptan fácilmente al trabajo en equipo. Son emprendedores, más líderes que gerentes por naturaleza.

Valoran la vida al aire libre, el contacto constante con la naturaleza para realizar deportes, siempre con una tendencia al desafío personal e interpersonal.

1.4.3. Generación Y (*millennials*)

Esta generación se conforma por las personas nacidas entre los años 1984 y 1996⁵. Se distingue por el uso (hasta un punto excesivo) de la tecnología de la información y comunicación. Conocida también por su búsqueda de libertad en todos sentidos.

Buscan la eficiencia para lograr los objetivos de los negocios, incluso pueden ser quienes orienten a *baby boomers* y generación X, por medio del uso

⁵ PEREZBOLDE, Guillermo. *Diferencias entre generaciones*. <https://www.merca20.com/conoce-las-diferencias-entre-millennials-genx-y-baby-boomers/3>. Consulta: octubre 2018.

inteligente de la tecnología, convirtiendo esta herramienta un factor indispensable.

En el aspecto laboral gustan ser emprendedores y multitareas, aprovechando siempre herramientas tecnológicas que les permitan tener trabajos incluso desde casa.

1.4.4. Generación Z (*centennials*)

Esta generación se conforma por las personas nacidas entre los años 1997 y 2010⁶. Una generación poco analizada en comparación con las anteriores, a la fecha quienes nacieron en el año 1997 recién cumplen o están por cumplir los 22 años, edad en la que se proyecta una fase intermedia o final de la etapa universitaria, esto denota poca actividad laboral formal.

Empresas y marcas tienen poco tiempo analizando a esta nueva generación, resultados como el de ManPower (2016) permiten proyectar a esta generación como 24 % de la fuerza laboral a nivel mundial⁷. Ese porcentaje refleja el impacto de nuevos profesionales egresados de universidades acoplándose a la fuerza laboral, lo cual crea la necesidad de conocerlos y, llevado ese conocimiento a la aplicación, crear un impacto positivo.

Esta generación es caracterizada por haber nacido en el auge de la innovación tecnológica mayormente en los dispositivos móviles, el uso de tales les resulta familiar como un idioma natal. Hacen uso de sus teléfonos

⁶ Infobae. *Características de generación centennial*. <https://www.infobae.com/especiales/2017/08/16/el-futuro-en-manos-de-la-nueva-juventud-millennial-y-centennial>. Consulta: octubre 2018.

⁷ Ibíd.

inteligentes un 30 % más que el resto de las generaciones⁸, pero a diferencia de la generación Y se enfocan en la facilidad de uso y de aplicaciones y sitios web. Respecto al uso de redes sociales, la tendencia son las redes sociales de comunicación más privada y directa.

Dado que no hay un estudio que revele comportamiento en el ámbito laboral, esta generación no será analizada en las posteriores secciones de esta investigación.

⁸ Estrategias y Negocios. *Centennials, el próximo gran reto para las marcas de consumo*. <https://www.estrategiaynegocios.net/empresasymangement/1239689-330/centennials-el-pr%C3%B3ximo-gran-reto-para-las-marcas-de-consumo>. Consulta: octubre 2018.

1.4.5. Comparación de tendencias e influencias tecnológicas

En la siguiente tabla se muestra la comparación de las tendencias tecnológicas que influenciaron las generaciones en análisis.

Tabla I. Comparación de tendencias e influencias tecnológicas

Baby Boomers	Generación X	Generación Y
Nacimiento de la televisión	Transición en el uso de aparatos eléctricos a uso de aparatos electrónicos	Nacen cuando el Internet está en un punto muy alto en uso y popularidad
Llegada del <i>fax</i>	Aumento de popularidad del Internet	SMS
Relojes digitales de pulsera	Uso del “.com”	Reproductores MP3, MP4, DVD
Nacimiento de la telefonía celular	Pacman	<i>Smartphones</i>
Computadora personal	Ipod	
Internet		
Llegada del hombre a la luna		
Transición en el cine en blanco y negro a color		

Fuente: elaboración propia.

1.4.6. Comparación de metodologías de su formación académica

La siguiente tabla muestra comparación de las metodologías de formación académica utilizadas en cada generación.

Tabla II. Comparación de metodologías de su formación académica

NIVEL EDUCATIVO	BABY BOOMERS	GENERACIÓN X	GENERACIÓN Y
NIVEL INTERMEDIO	Educados por maestros tradicionalistas, muy estrictos, con valores morales como base.	Formados con el uso de computadoras personales e Internet	Educados por medio de clases tradicionales, pero con interacción constante con la tecnología informática
EDUCACIÓN SUPERIOR	A penas experimentaron material educativo en televisión, poca interacción con las computadoras personales y el Internet.	Equipados con buenas herramientas, no se limitan a estudiar solo la universidad, buscan maestrías y doctorados.	Acompañan la formación universitaria tradicional con autoayuda, haciendo uso de tutoriales en línea, experimentación con aplicaciones móviles, etc.

Fuente: elaboración propia.

2. ANÁLISIS DE SITUACIÓN ACTUAL Y PROPUESTA DE SOLUCIÓN

2.1. Definición del problema

En Guatemala se ha notado un crecimiento significativo en la necesidad de la automatización de procesos y procedimientos para el manejo de información en todos los segmentos de mercado. El desarrollo de software es gran parte la solución a esta necesidad, por tanto, la demanda ha aumentado. La demanda es cubierta por empresas privadas y empleos en modalidad *freelance*. Los involucrados en los proyectos de desarrollo de software pertenecen a distintas generaciones, esto implica diferencias en criterios de decisión, formalismo para trabajar y las metodologías que se eligen para cumplir las entregas de proyectos como soluciones.

No hay ningún estudio que relacione el análisis generacional con los criterios de decisión y aceptación de los líderes de proyectos de desarrollo, esto implica que los interesados de los proyectos de desarrollo de software no cuentan con una herramienta que les permita especializarse para atender a la necesidad con mayor experiencia, sino tener conocimientos generales de todas las metodologías, lo cual puede dejar puntos débiles en su gestión del proyecto y propiciar una mala planificación de tiempo y recursos.

2.2. Propuesta de solución al problema

Proporcionar una herramienta que ayude a los interesados de los proyectos de desarrollo de software a identificar las metodologías que más se usan y las razones por las cuales son elegidas, tomando como base la generación de los líderes de proyectos que toman decisiones. La herramienta será el análisis de los resultados de una encuesta realizada a líderes e interesados de proyectos de desarrollo de software, con un enfoque al análisis generacional y principales criterios de decisión.

2.2.1. Estructura de encuesta y detalle del objetivo de cada elemento

- ¿Cuál es su fecha de nacimiento?
Objetivo: posicionar a la persona encuestada en una de las generaciones analizadas.
- Fecha de graduación o ingreso de estudios universitarios.
Objetivo: identificar la etapa de formación universitaria, segmentada por encuestados graduados y no graduados a nivel universitario.
- Rol como interesado en el desarrollo de software.
Opciones: gerente de proyecto, *project manager*, proveedor/consultor, analista y programador.
Objetivo: delimitar e identificar el rol de la persona encuestada en proyectos de desarrollo de software.
- ¿Qué tipo de mercado objetivo tienen los proyectos de desarrollo en los que está trabajando?
Opciones: financiero, gubernamental, responsabilidad social, educativo, publicidad, entretenimiento, *e-commerce*, otro.

Objetivo: delimitar e identificar el segmento de mercado atendido en los proyectos en los que trabaja la persona encuestada.

- ¿Qué metodologías de desarrollo de software le fueron enseñadas en su etapa universitaria?

Opciones: modelo en cascada, modelo en espiral, metodología de prototipos, Scrum, Extreme Programming, Kanban, otra.

Objetivo: delimitar e identificar las principales metodologías de desarrollo de software enseñadas en la etapa universitaria.

- ¿Usted trabaja?

Opciones: Sí o No.

Objetivo: segmentar la población objetivo.

- ¿Qué metodologías de desarrollo de software considera mejores para el mercado guatemalteco?

Opciones: modelo en cascada, modelo en espiral, metodología de prototipos, Scrum, Extreme Programming, Kanban, otra.

Objetivo: identificar las metodologías preferidas por estudiantes de carreras universitarias afines a ciencias de la información.

- ¿Qué metodologías de desarrollo de software ha usado (hasta un mes atrás)?

Opciones: modelo en cascada, modelo en espiral, metodología de prototipos, Scrum, Extreme Programming, Kanban, otra.

Objetivo: identificar las metodologías de desarrollo utilizadas por la persona encuestada a través de su tiempo de experiencia.

- Respecto a las metodologías seleccionadas previamente, ¿qué ventajas ha identificado al trabajar con ellas?

Objetivo: establecer la importancia y valor que las metodologías han aportado a la persona encuestada.

- ¿Qué metodologías de desarrollo de software usa actualmente?

Opciones: modelo en cascada, modelo en espiral, metodología de

prototipos, Scrum, Extreme Programming, Kanban, otra.

Objetivo: identificar las metodologías de desarrollo de software utilizadas actualmente.

- Respecto a las metodologías seleccionadas previamente, ¿qué ventajas ha identificado ante las demás?

Objetivo: identificar la relación que puede existir entre las ventajas identificadas en las metodologías usadas actualmente y las usadas en el pasado.

- ¿Qué tipo de entregables prefiere en las etapas de desarrollo de software?

Objetivo: obtener información específica acerca de los criterios de aceptación de la persona encuestada.

3. ANÁLISIS DE RESULTADOS

En este capítulo se muestran los resultados obtenidos en las encuestas realizadas. Al sector económicamente activo se realizó una entrevista para obtener más información de su experiencia profesional. Además se presenta el análisis de resultados.

Se realizaron 34 encuestas con la herramienta Google Forms, de las cuales 31 se extendieron a entrevista, esto correspondiente a la población económicamente activa. Es importante resaltar la falta de participación de la generación Baby Boomers para responder las encuestas, por tanto, en la tabulación de los datos y tablas comparativas se toman en cuenta solamente las generaciones “X” y “Y”.

3.1. Tabulación de datos

En esta sección se muestran los datos obtenidos en el orden que fue estructurada la encuesta.

3.1.1. Resultados obtenidos con respecto a la edad de los encuestados

Las fechas de nacimiento se presentan en listados a dos columnas, la primera columna agrupa mes y año, y la segunda detalla el día.

Figura 3. Resultados. Fecha de nacimiento

¿Cuál es su fecha de nacimiento?

34 respuestas

feb. 1980	1	sept. 1987	10	nov. 1991	12	13
sept. 1980	10	may. 1988	26	may. 1992	17	
may. 1981	24	nov. 1988	15	oct. 1992	29	
ene. 1984	11	jun. 1989	25	nov. 1992	7	
ene. 1985	11	jul. 1989	4	ene. 1993	11	
feb. 1985	15	oct. 1989	16	dic. 1993	6	
sept. 1985	24	abr. 1990	28	ene. 1995	9	
feb. 1986	21	jun. 1990	8	jun. 1995	3	
abr. 1986	5	jul. 1990	28	dic. 1995	1	
oct. 1986	10	ago. 1990	30			
jul. 1987	25	jul. 1991	19			

Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Los resultados obtenidos denotan un rango de fechas de nacimiento del 01/02/1980 al 01/12/1995. Este resultado respalda el análisis completo de las

generaciones excepto *centennials* (Z), debido a la ausencia en el mercado laboral y estudiantil universitario en fase final.

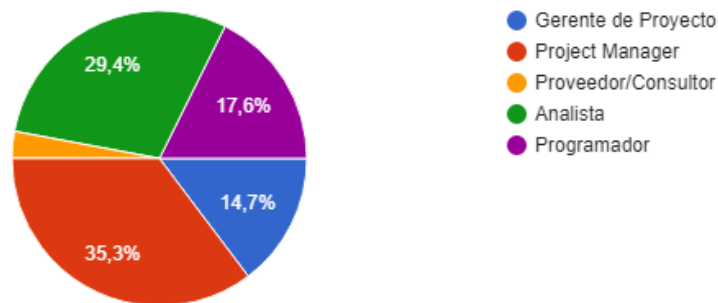
3.1.2. Resultados obtenidos con respecto al rol de participación de los entrevistados en los proyectos que ejecutan

Los datos de la participación que tienen los encuestados en los proyectos que ejecutan se presentan como porcentajes en una gráfica de pastel.

Figura 4. Resultados. Rol como interesado en el proyecto

Rol como interesado en el proyecto de desarrollo de software

34 respuestas



Nota: El conteo de Proveedor/Consultor es 1(2,9%), el dato no se muestra en la imagen debido a la dimensión del porcentaje respecto a los demás roles.

Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

El resultado obtenido denota mayor porcentaje de *project manager*, siendo el principal público objetivo y de análisis en esta investigación por la responsabilidad de elección de las distintas metodologías, seguido de analistas, quienes acompañan la ejecución de actividades administrativas y analíticas que les requieren. El tercer rol en el que más participan corresponde a

programadores, quienes ejecutan las tareas técnicas. El cuarto rol en el que más se participa representa a gerentes de proyecto, quienes en las metodologías son encargados de priorizar los requerimientos y actividades durante el tiempo de vida del proyecto. Cada proyecto cuenta regularmente con un solo gerente, por lo que el porcentaje obtenido es significativo en el mercado laboral.

El rol con menor participación corresponde a proveedor/consultor, rol que puede tomar responsabilidades de gerente de proyecto y de *project manager* en actividades administrativas como consultor. Como proveedor toma responsabilidades administrativas de *project manager* y técnicas de programador.

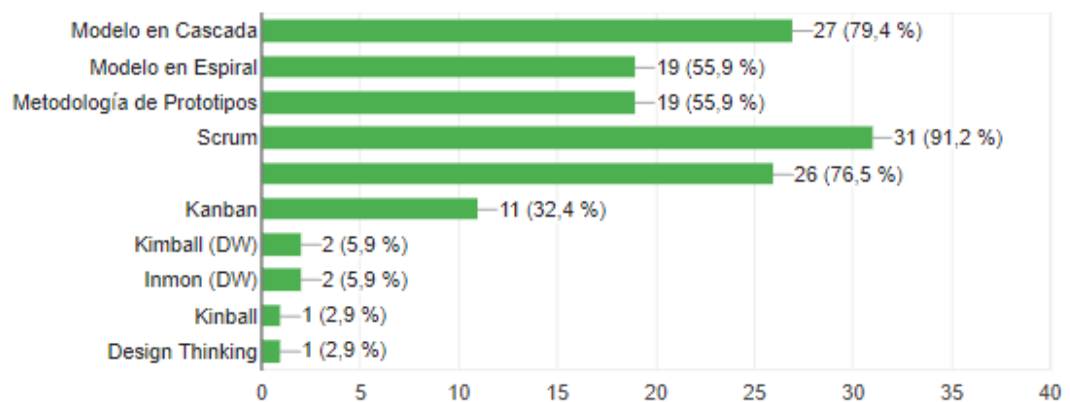
3.1.3. Metodologías de desarrollo en la etapa universitaria

Los datos de las metodologías aprendidas por los encuestados en su formación universitaria se presentan en una gráfica de barras porcentuales de forma horizontal.

Figura 5. **Resultados. Metodologías de desarrollo aprendidas en etapa universitaria**

¿Qué metodologías de Desarrollo de Software le fueron enseñadas en su etapa universitaria?

34 respuestas



Nota: El conteo 26 (76,5%) corresponde a metodología Extreme Programming (Programación Extrema -XP), por el título usado en la encuesta no se muestra la etiqueta.

Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Este resultado denota el auge de la metodología Scrum con el mayor porcentaje y la permanencia de las metodologías clásicas en la formación universitaria.

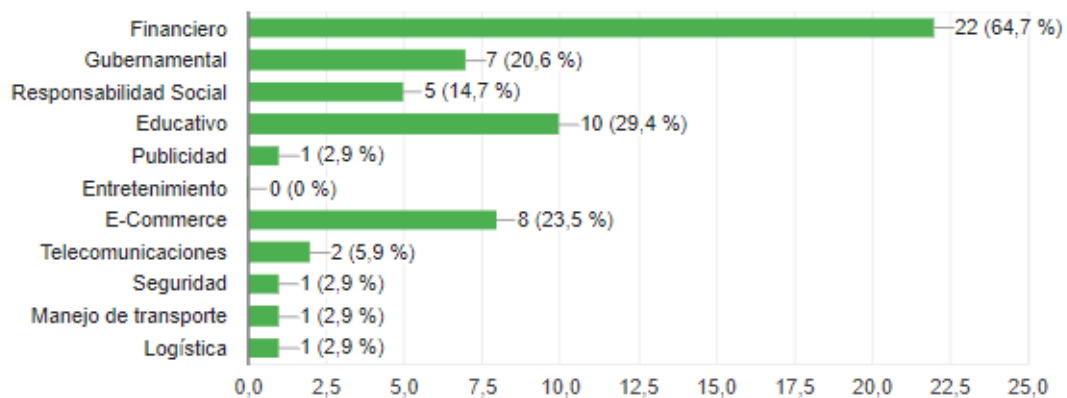
3.1.4. Mercado objetivo de los proyectos en los que labora

Los datos del mercado objetivo que tienen los proyectos en los que se desenvuelven los encuestados, se presentan en una gráfica de barras porcentuales de forma horizontal.

Figura 6. **Resultados. Mercado objetivo de los proyectos en los que trabaja**

¿Qué tipo de mercado objetivo tienen los proyectos de desarrollo de software en los que está trabajando?

34 respuestas



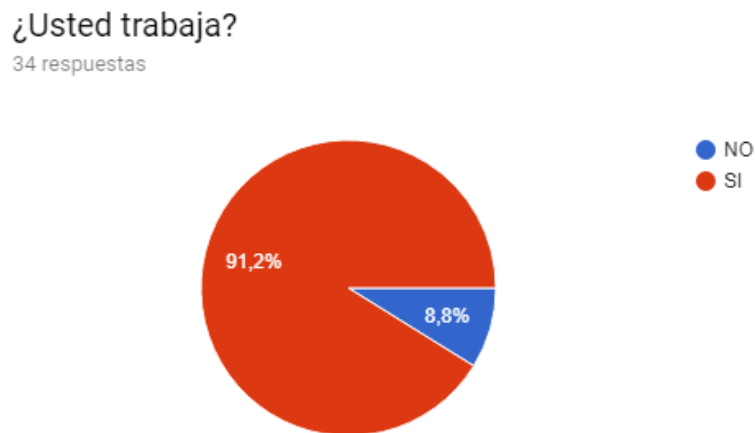
Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Este resultado muestra la amplitud del mercado laboral tecnológico en el sector financiero, esto se debe a la necesidad de automatización de sus procesos con el objetivo de optimizar tiempo de atención a clientes.

3.1.5. Población económicamente activa

Los datos de la participación de los encuestados en el mercado laboral, se presenta en una gráfica de pastel.

Figura 7. **Resultados. Segmentación de población económicamente activa**



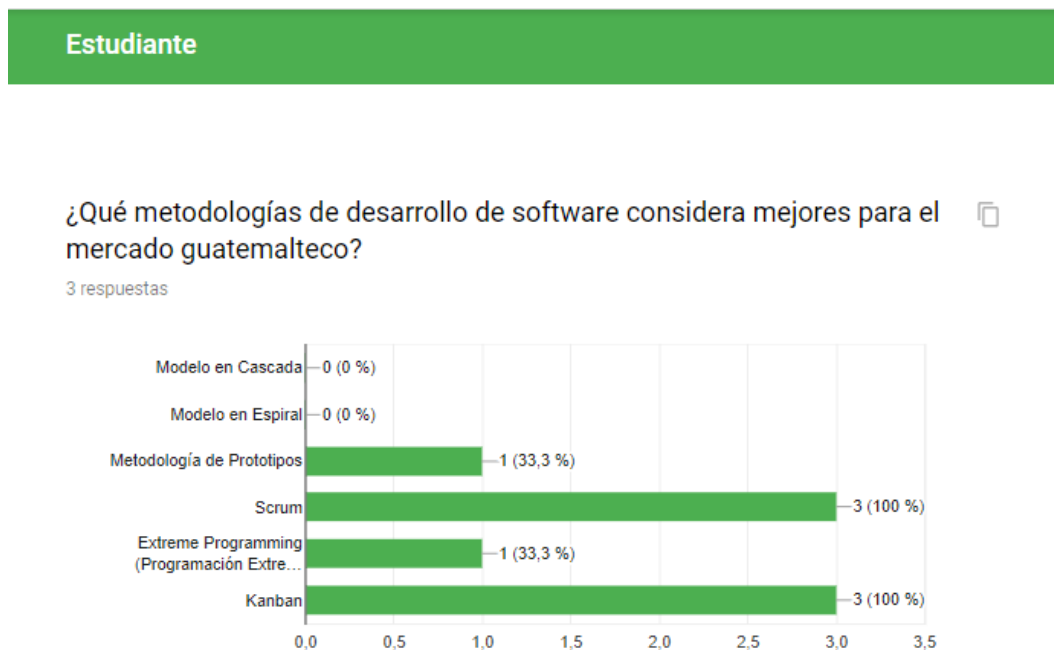
Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Este resultado denota la alta presencia de la población encuestada en el mercado laboral. El 8,8 % que no labora se encuentra en fase final de estudios universitarios, esto tomando en cuenta los cursos relacionados a las metodologías de desarrollo de software.

3.1.6. Metodologías consideradas idóneas para el mercado guatemalteco en estudiantes

Los datos de las metodologías que los encuestados consideran idóneas para el mercado guatemalteco, se presentan en una gráfica de barras porcentuales de forma horizontal.

Figura 8. **Resultados. Metodologías idóneas para el mercado guatemalteco**



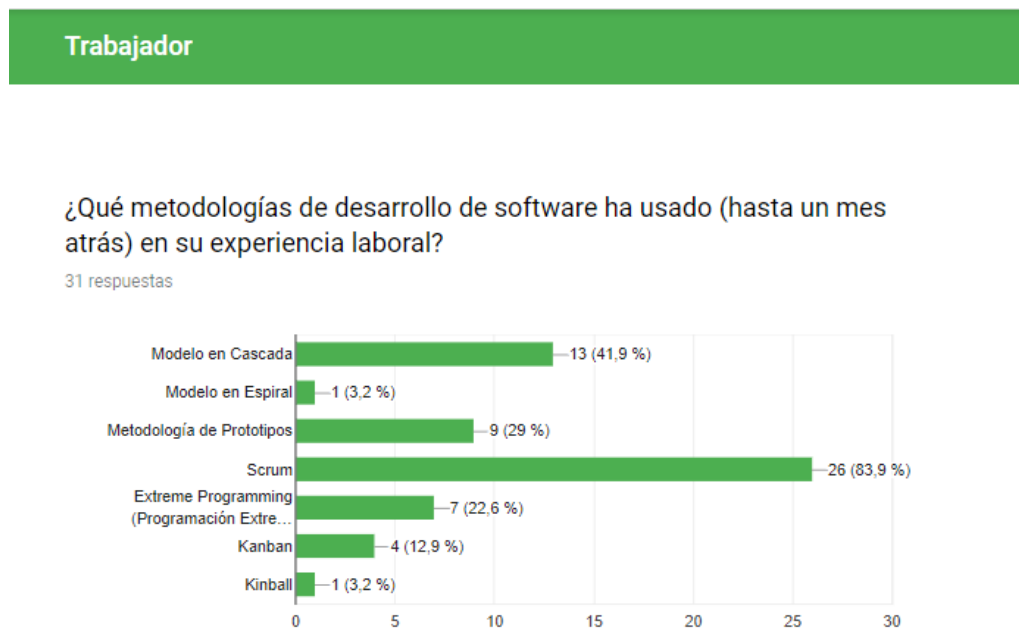
Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Este resultado se obtuvo de la población encuestada que aún no es activa en el mercado laboral. Su decisión es acorde a las metodologías ágiles que están en auge, pero hay un desconocimiento de los procesos de un proyecto que no involucran el desarrollo de software, estos procesos sí están contenidos en las metodologías clásicas.

3.1.7. Metodologías utilizadas actualmente en el mercado laboral un mes atrás

Los datos de las metodologías utilizadas un mes atrás, con base en la fecha de la encuesta, se presentan en una gráfica de barras porcentuales de forma horizontal.

Figura 9. **Resultados. Metodologías usadas hasta un mes atrás de la encuesta**



Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

Este resultado muestra el mayor porcentaje de uso de la metodología ágil Scrum, la segunda más usada es la metodología clásica de modelo en cascada. Esto marca un punto de partida en el análisis de la preferencia de la población encuestada por las distintas metodologías.

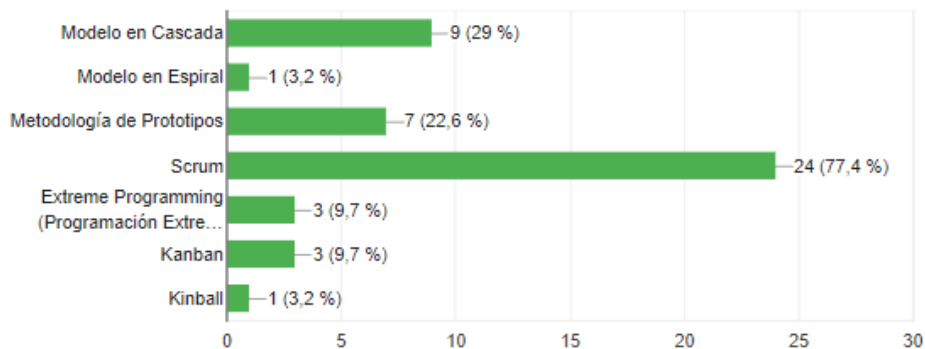
3.1.8. Metodologías utilizadas a la fecha en el mercado laboral

Los datos de las metodologías utilizadas por los encuestados en la fecha de la encuesta se presentan en una gráfica de barras porcentuales de forma horizontal.

Figura 10. **Resultados. Metodologías usadas en la fecha de encuesta**

¿Qué metodologías de desarrollo de software usa actualmente?

31 respuestas



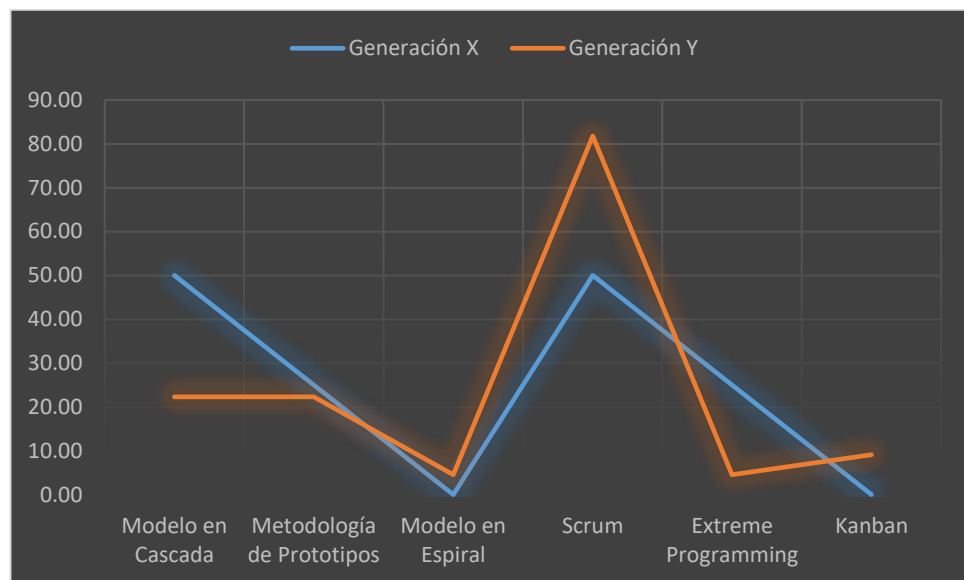
Fuente: estadísticas de encuesta elaborada, empleando Google Forms.

En este resultado se aprecia que Scrum es la metodología más utilizada y la segunda en porcentaje es la metodología clásica de modelo en cascada, resultado apegado al dato de un mes previo. Esta tendencia da oportunidad para resaltar la estrecha relación que existe entre estas dos metodologías, pues la metodología clásica es el fundamento de la ágil. La población encuestada utiliza modelo en cascada para la gestión del proyecto en general y Scrum para el desarrollo de software específicamente. Esta combinación ha generado mayor satisfacción a los gerentes de proyectos pues tiene una documentación formal y resultados más rápidos comparados a la ejecución de una metodología clásica únicamente.

3.2. Clasificación de líderes según su criterio de decisión

La clasificación de las metodologías, criterios de decisión y generación de los líderes de proyectos, se presentan en dos gráficas de doble eje y una tabla.

Figura 11. **Clasificación de metodologías elegidas por generación (cifra porcentual)**

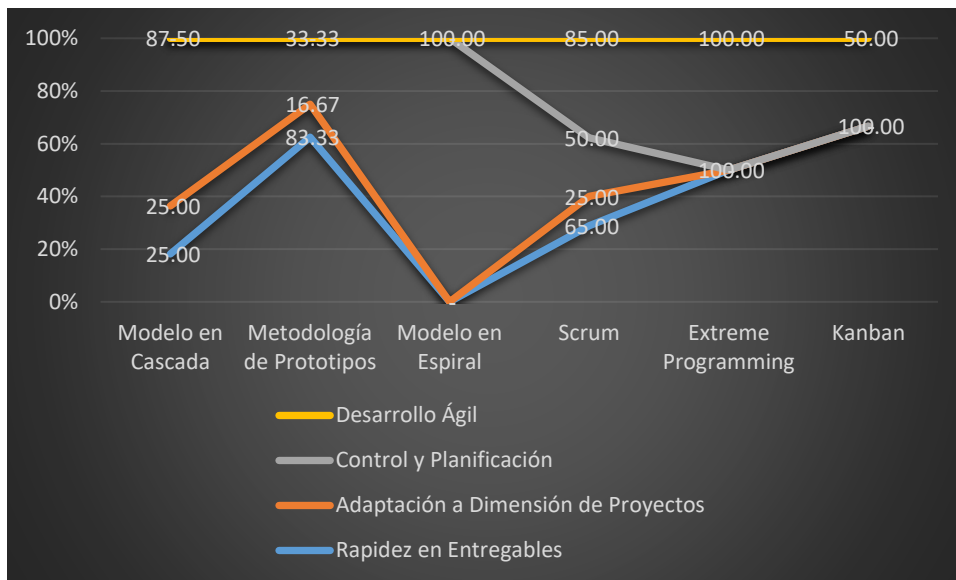


Fuente: elaboración propia, empleando MS Excel.

En la figura 11 se aprecia la preferencia de la generación Y por la metodología ágil Scrum y la generación X por las metodologías clásicas, sin dejar por un lado el uso de Scrum. Las personas encuestadas que pertenecen a la generación X ocupan cargos gerenciales y como líderes de proyectos. Tienen responsabilidad en todo el tiempo de vida de un proyecto, debido a eso eligen las metodologías por el mayor alcance y fundamentos legales que estas tienen.

Las personas encuestadas que pertenecen a la generación Y tienen poco tiempo en el mercado laboral, ocupan cargo de programadores y analistas, están involucradas en actividades propias del desarrollo de software.

Figura 12. **Características preferidas de las metodologías (cifra porcentual)**

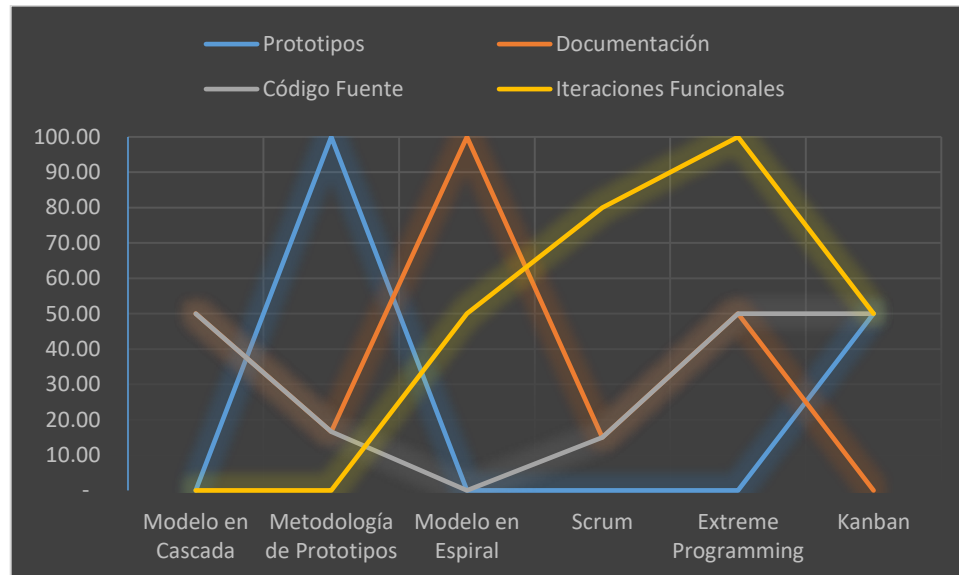


Fuente: elaboración propia, empleando MS Excel.

En la figura 12 se aprecia por cuáles características los individuos prefieren utilizar cada una de las metodologías. En el caso de la metodología Scrum las características por las que más se utiliza es por el desarrollo ágil y rapidez en entregables, características propias de la fase de desarrollo de software en un proyecto, es por ello que se concluye que la generación Y prefiere Scrum debido a los roles que ocupa actualmente, lo cual los lleva a visualizar los proyectos desde una perspectiva no formal.

En el caso de las metodologías clásicas de modelo en cascada y prototipos los individuos las prefieren por sus características de control y planificación, y adaptación a la dimensión de los proyectos. Si se consideran estas características y que son las metodologías que prefieren los individuos de la generación X es posible concluir que es porque tienen una visión completa en todo el ciclo de vida del proyecto.

Figura 13. **Criterios de aceptación por metodología (cifra porcentual)**



Fuente: elaboración propia, empleando MS Excel.

En la figura 13 se muestran las iteraciones funcionales como principal criterio de aceptación para las metodologías ágiles, y la documentación y prototipos como principales criterios de aceptación para las metodologías clásicas.

La tabla III muestra en resumen los datos resultantes de las figuras 11, 12 y 13.

Tabla III. **Clasificación de líderes según su generación y criterio de decisión**

	GENERACIÓN X	GENERACIÓN Y
METODOLOGÍAS DE SU PREFERENCIA (EN ORDEN DE PRIORIDAD)	Modelo en cascada Metodología de prototipos Scrum Modelo en espiral	Scrum Modelo en cascada Prototipos Kanban Extreme Programming
ENTREGABLES DE SU PREFERENCIA (EN ORDEN DE PRIORIDAD)	Prototipos funcionales Prototipos no funcionales Iteraciones funcionales Código fuente Documentación formal	Iteraciones funcionales Prototipos funcionales

Fuente: elaboración propia.

3.3. **Análisis de resultados por generación**

3.3.1. **Generación X**

A esta generación le fueron enseñadas las metodologías clásicas como base principal y las metodologías ágiles en muy pocos casos. De allí parte su preferencia por los prototipos (funcionales y no funcionales), y la necesidad de requerir código fuente y documentación formal en los entregables preliminares, como se muestra en figura 13, tomando las metodologías clasificadas en la tabla III. Gran parte de los encuestados de esta generación ha establecido como características principales el control, la planificación y la adaptación a la dimensión de proyectos, como se muestra en la figura 12.

A esto se agrega una tendencia importante en su criterio de decisión: las iteraciones funcionales. Estas son propias implementaciones cortas del modelo

en cascada, que en la actualidad se expande con las buenas prácticas de Scrum. Se muestra en la figura 11 con preferencia en esta metodología por parte de la Generación X.

Su estrecha relación con profesionales de la generación Y abre su aceptación, sin dejar atrás entregables que sean visuales preliminares de la entrega final, a diferencia de las entregas modulares como partes atómicas de un sistema.

3.3.2. Generación Y

A esta generación también le fueron enseñadas las metodologías clásicas como fundamento, sin embargo, el mayor énfasis se dio a las metodologías ágiles en cursos de la fase final de la carrera universitaria. De forma específica se puede citar a los encuestados que nacieron entre 1985 y 1992 con criterios muy similares a la Generación X, de este punto nace la tendencia al uso de metodologías clásicas como se muestra en la figura 11.

La metodología preferida en su mayoría es Scrum, con su principal entregable, iteración funcional, tal como se muestra en la figura 13. Es preferida debido al orden, rapidez (como se muestra en la figura 12) y ausencia de documentación exhaustiva para las distintas partes interesadas.

CONCLUSIONES

1. Sí existe una estrecha relación entre las metodologías de desarrollo de software que los líderes de proyectos eligen y la generación a la que pertenecen, su etapa de formación influenciada por la tecnología innovadora en su momento y el formalismo propio de su generación.
2. Se identificaron los principales entregables de las distintas metodologías de desarrollo de software y se clasificaron según su etapa en el ciclo de vida del desarrollo.
3. Se identificaron tendencias en la elección de metodologías ágiles, las cuales se basan en principios de metodologías clásicas.
4. Los criterios de decisión de la Generación X son control y planificación y adaptación a la dimensión de los proyectos. Los criterios de decisión de la Generación Y son el desarrollo ágil y la rapidez en entregables.

RECOMENDACIONES

1. A los líderes de proyectos de desarrollo de software que utilizan solamente metodologías clásicas, se recomienda identificar metodologías ágiles que tengan como base los principios de las metodologías clásicas, lo cual ayudará a reducir tiempos en documentación exhaustiva, sin perder las ventajas de las etapas de ciclo de vida de software correspondientes.
2. Tomar en cuenta los entregables preferidos por los líderes de proyectos y, a partir de ello, buscar especialización en las metodologías que tienen dichos entregables.
3. Ampliar el contenido de las metodologías ágiles, como partes interesadas, ventajas y desventajas. Estos aspectos no agregaban valor a esta investigación, sin embargo, agregarán valor al implementarlos.
4. A los líderes de proyectos que cuentan con equipos de trabajo con integrantes que pertenecen a la Generación Y, se recomienda transmitir conocimiento de la gestión de los proyectos desde una perspectiva más formal, la cual es proporcionada por el uso de metodologías clásicas.

BIBLIOGRAFÍA

1. ALONSO AMO, Fernando; MARTÍNEZ NORMAND, Loic; SEGOVIA PÉREZ, Francisco Javier. *Introducción a la ingeniería del software modelos de desarrollo de programas*. España: Delta Publicaciones, 2005.
2. Estrategias y Negocios. *Tendencias de la Generación Centennial. Centennials, el próximo gran reto para las marcas de consumo*. [en línea].
<<https://www.estrategiaynegocios.net/empresasmanagement/1239689-330/centennials-el-pr%C3%B3ximo-gran-reto-para-las-marcas-de-consumo>>. [Consulta: 2019].
3. GARCÍA FLORES, Jacinto; FUENTES ROJAS, José Arturo; LÓPEZ MORENO, María; SILVA AMBRIZ, Linda Laura; CAJICA ÁNGELES, Edith; FLORES HERNÁNDEZ, María Luisa. *Estrategias de enseñanza aprendizaje a utilizar en las generaciones Baby Boomer, X, Y y Z en la educación superior*. [en línea]. <<https://www.uv.mx/icp/files/2018/02/Num05-Art010.pdf>>. [Consulta: octubre 2018].
4. Gerza. *Estilo de aprendizaje de la Generación X*. [en línea]. <http://www.gerza.com/articulos/aprendizaje/todos_articulos/estilo_aprendiz_gener_x.html>. [Consulta: octubre 2018].

5. GHAAHRAI, Amir. *What is prototyping*. [en línea]. <<https://www.testingexcellence.com/prototyping-model-software-development/>>. [Consulta: octubre 2018].
6. GURENDO, Dmitry. *Software development life cycle. Spiral Model*. [en línea]. <<https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/>>. [Consulta: octubre 2018].
7. Infobae. *Características de generación Centennial. El futuro en manos de la nueva juventud millennial y centennial*. [en línea]. <<https://www.infobae.com/especiales/2017/08/16/el-futuro-en-manos-de-la-nueva-juventud-millennial-y-centennial/>>. [Consulta: octubre 2018].
8. KPMG International Cooperative. *El desafío de retener a distintas Generaciones*. [en línea]. <<https://assets.kpmg.com/content/dam/kpmg/pa/pdf/delineandoestrategias/DE-El-desafio-de-retener-generaciones.pdf>>. [Consulta: octubre 2018].
9. MARTINEZ, Julian. *Estilo de aprendizaje de la Generación Millennials. Los millennials, la generación que reta a la educación*. [en línea]. <<https://revistaeducacionvirtual.com/archives/2188>>. [Consulta: octubre 2018].

10. MEDINILLA, A., GINÉ, A., GÓMEZ, E. *Principios del manifiesto ágil. Manifiesto por desarrollo ágil de software*. [en línea]. <<http://agilemanifesto.org/iso/es/principles.html>>. [Consulta: octubre 2018].
11. MEGINO BARQUINERO, José María. *Qué es un contrato de operaciones*. [en línea]. <<https://www.seas.es/blog/informatica/analisis-y-diseno-de-software-contratos-de-operaciones/>>. [Consulta: noviembre 2018].
12. NAVARRO, Sandra. *Tipos de contratos. Cómo hacer la mejor elección. Contratos para proyectos de desarrollo de software*. (2014) [en línea]. <https://www.cursodireccionproyectos.com/2014/07/tipos-de-contratos-y-como-hacer-la-mejor-eleccion/>. [Consulta: noviembre 2018].
13. Nielsen, N.V. *Estilos de vida generacionales*. [en línea]. <<https://www.nielsen.com/content/dam/nielsen-global/latam/docs/reports/2016/EstilosdeVidaGeneracionales.pdf>>. [Consulta: octubre 2018].
14. OBS. Metodologías de Desarrollo de Software. *Modelo de Cascada*. [en línea]. <<https://www.obs-edu.com/int/blog-project-management/metodologia-agile/pros-y-contras-de-la-metodologia-en-cascada>>. [Consulta: octubre 2018].

15. PEREZBOLDE, Guillermo. *Diferencias entre generaciones. Conoce las diferencias entre millennials, GenX y baby boomers*. [en línea]. <<https://www.merca20.com/conoce-las-diferencias-entre-millennials-genx-y-baby-boomers/3/>>. [Consulta: octubre 2018].
16. Project Management Institute. *Agile practice guide*. Pennsylvania: Agile Alliance, 2017.
17. Proyectos ágiles. *Historia de Scrum – Proyectos Ágiles*. [en línea]. <<https://proyectosagiles.org/historia-de-scrum/>>. [Consulta: octubre 2018].
18. Proyectos ágiles. *Qué es Scrum*. [en línea]. <<https://proyectosagiles.org/que-es-scrum/>>. [Consulta: octubre 2018].
19. SMITS, Hubert. *¿Cómo hacer contratos para proyectos ágiles?* [en línea]. <<https://sg.com.mx/revista/%C2%BFc%C3%B3mo-hacer-contratos-para-proyectos-%C3%A1giles>>. [Consulta: octubre 2018].
20. SOMMERVILLE, Ian. *Ingeniería del software*. Madrid: PEARSON EDUCACION, S.A., 2005.
21. WELLS, Don. *The rules of Extreme Programming*. [en línea]. <<http://www.extremeprogramming.org/>>. [Consulta: octubre 2018].